УДК 519.161

О.Б. Маций[1], *А.В. Морозов*[2], *А.В. Панишев*[2]
[1]Харьковский национальный автомобильно-дорожный университет
Украина, 61002, г. Харьков, ул. Петровского, 25
[2]Житомирский государственный технологический университет
Украина, 10005, г. Житомир, ул. Черняховского, 103

Рекуррентный метод решения задачи о назначениях

O.B. Matsiy[1], *A.V. Morozov*[2], *A.V. Panishev*[2]
[1]Kharkiv National Automobile and Highway University, Ukraine
Ukraine, 61002, c. Kharkiv, Petrosvkogo st., 25
[2]Zhytomyr State Technological University
Ukraine, 10005, c. Zhytomyr, Chernyakhivskogo st., 103

Recurrent Method for Solving The Assignment Problem

О.Б. Мацій[1], *А.В. Морозов*[2], *А.В. Панішев*[2]
[1]Харківський національний автомобільно-дорожній університет
Україна, 61002, м. Харків, вул. Петровського, 25
[2]Житомирський державний технологічний університет
Україна, 10005, м. Житомир, вул. Черняхівського, 103

Рекурентний метод розв'язання задачі про призначення

В статье предлагается новый метод решения задачи о назначениях, основанный на рекурсивном получении оптимального решения задачи. Он состоит в нахождении взвешенного паросочетания минимального суммарного веса в двудольном графе, используя понятия кратчайшего увеличивающего пути. Предложенный метод позволяет получать решения задачи о назначениях значительно быстрее, чем существующие методы.
Ключевые слова: задача о назначениях, паросочетание, двудольный граф, увеличивающий путь.

The article proposes new method for solving the assignment problem based on recursive obtaining an optimal solution. It consists in finding a minimum weighted matchings total weight in a bipartite graph, using concepts the shortest increasing path. The proposed method allows to obtain solutions of the assignment problem is significantly faster than existing methods.
Key words: the assignment problem, matching, bipartite graph, increasing path.

У статті пропонується новий метод розв'язання задачі про призначення, який ґрунтується на рекурсивному отриманні оптимального розв'язку задачі. Він полягає в знаходженні зваженого паросполучення мінімального сумарної ваги в дводольному графі, використовуючи поняття найкоротшого збільшуючого шляху. Запропонований метод дозволяє отримувати розв'язки задачі про призначення значно швидше, ніж існуючі методи.
Ключові слова: задача про призначення, паросполучення, дводольний граф, збільшуючий шлях.

Широко известные методы решения задачи о назначениях (ЗН), такие, как венгерский метод, метод Кана-Мункреса и метод потенциалов, построены с использованием разных подходов, применяемых в комбинаторной оптимизации, и характеризуются разной временной сложностью, не меньшей, чем $O\left(n^3\right)$, где n – порядок матрицы стоимостей [1]. В [2] изложен алгоритм решения одного из вариантов ЗН, стоимость которого понижена до $O\left(n^2\right)$. В [2] показано, что он выполняет функции процедуры, встроенной в метод ветвей и границ для быстрого вычисления более точных нижних оценок стоимости

замкнутых маршрутов в задаче коммивояжера. Алгоритм состоит в нахождении взвешенного паросочетания минимального суммарного веса в двудольном графе с $2n$ вершинами, используя введенные в [2] понятия кратчайшего увеличивающего пути. В данной статье описан рекуррентный метод решения ЗН, развивающий результаты работ [2], [3] и технически упрощающий наиболее распространенный венгерский метод.

Перестановочно-матричная постановка задачи. Опишем рассматриваемый здесь алгоритм решения ЗН, исходя из её формулировки в следующем виде.

Для матрицы стоимостей (весов) $C=\left[c_{ij}\right]_n$, где $c_{ij}\in R_0^+$ или $c_{ij}=\infty$, R_0^+ – множество неотрицательных действительных чисел, найти

$$C(\sigma)=\min_{\pi}\sum_{i=1}^{n}c_{\pi[i]}. \quad (1)$$

Здесь $\pi=(\pi[1],\ \pi[2],\ ...,\ \pi[n])$ – перестановка множества $\{1,\ 2,\ \ldots,\ n\}$ номеров столбцов матрицы C, $\sigma=(\sigma[1],\ \sigma[2],\ ...,\ \sigma[n])$ – оптимальная перестановка стоимостью $C(\sigma)=\sum_{i=1}^{n}c_{\sigma[i]}$, $c_{\sigma[i]}\neq\infty$, $i=\overline{1,n}$, или решение ЗН. Перестановку $\pi=(\pi[1],\ \pi[2],\ ...,\ \pi[n])$, в которой $c_{\pi[i]}\neq\infty$, $i=\overline{1,n}$, назовем допустимым решением ЗН. Заметим, что ЗН с матрицей стоимостей, содержащей элементы $c_{ij}=\infty$, может не иметь решения. В этом случае необходимо установить, что множество допустимых решений задачи пусто. Будем искать σ, пошагово увеличивая на единицу число элементов k, $k=\overline{1,n}$, последовательности, образующей определенную часть допустимого решения ЗН. Рассмотрим свойства этой последовательности и способ её построения.

Любая часть допустимого решения ЗН, включающая k элементов, однозначно определяет подматрицу $\left[c_{i_s j_t}\right]_k$ матрицы C, $i_1<i_2<...<i_s<...<i_k$, $j_1<j_2<...$ $...<j_t<...<j_k$. Пусть последовательность $\pi_k=(\pi_k[i_1],\ \pi_k[i_2],\ ...,\ \pi_k[i_s],\ ...,\ \pi_k[i_k])$, $\pi_k[i_s]\in\{j_1,\ j_2,\ ...,\ j_t,\ ...,\ j_k\}$, $k=\overline{1,n-1}$, а) является решением ЗН для подматрицы $\left[c_{i_s j_t}\right]_k$, б) стоимость π_k не больше стоимости решения ЗН для любой подматрицы порядка k матрицы C. Если существует эффективная процедура преобразования последовательности π_k в последовательность $\pi_{k+1}, k=\overline{0,n-1}$, и задача (1) имеет решение, то на нахождение $\sigma=\pi_n$ требуется n шагов.

Матричный подход к построению оптимального назначения. Покажем, как строятся последовательности $\pi_k, k=\overline{1,n}$. Исходная последовательность $\pi_1=(\pi_1[i_1])$ определяется тривиально: в матрице C находится $c_{lr}=\min\left\{c_{ij}\mid i\leq i, j\leq n\right\}$, и, следовательно, $i_1=1$, $\pi[i_1]=r$.

Чтобы получить $\pi_2=(\pi_2[i_1],\ \pi_2[i_2])$, определим $c_{ms}=\min\left\{c_{ij}\mid i\neq l,\ j\neq r\right\}$, $c_{lp}=\ \min\left\{c_{lj}\mid j\neq r\right\}$, $c_{vr}=\min\left\{c_{ir}\mid i\neq l\right\}$ (рис. 1).

	c_{lp}		c_{lr}		
					c_{ms}
			c_{vr}		

Рисунок 1

Нетрудно видеть, что если $c_{lr} + c_{ms} \le c_{lp} + c_{vr}$, то условия а) и б) выполняются для последовательности π_2, в которой $i_1 = l,\ \pi_2[i_1] = r,\ i_2 = m,\ \pi_2[i_2] = s.$ Ей соответствует подматрица

c_{lr}	
	c_{ms}

.

В противном случае этим условиям удовлетворяет последовательность π_2 с элементами $\pi_2[i_1] = p,\ i_1 = l,\ \pi_2[i_2] = r,\ i_2 = v$ и подматрицей

c_{lp}	c_{lr}
	c_{vr}

.

Преобразуем последовательность π_2 в последовательность $\pi_3 = (\pi_3[i_1],\ \pi_3[i_2],\ \pi_3[i_3])$. Способ построения π_3 не зависит от того, какие элементы образуют π_2. Для определенности предположим, что $\pi_2 = (\pi_2[l] = r,\ \pi_2[m] = s)$.

Найдем $c_{wq} = \ = \min\left\{c_{ij} \mid i \ne l, m;\ \ j \ne s, r\right\}$ и $MIN1 = c_{lr} + c_{ms} + c_{wq}$.

Заметим, что $c_{wq} = c_{ms}$, если $\pi_2 = (\pi_2[l] = p,\ \pi_2[v] = r)$.

Преобразование π_2 в π_3 является результатом решения следующей вспомогательной задачи.

Для строк *l*, *m* и столбцов *r*, *s*, определяемых числами c_{lr} и c_{ms} матрицы *C*, требуется найти тройку элементов с минимальной суммой их значений при условии, что любые два элемента из тройки должны находиться в трех разных строках, включающих *l, m,* и трех разных столбцах, включающих *r, s*.

Если искомая тройка не содержит c_{lr}, но содержит c_{ms}, то сумма значений её элементов ограничена снизу величиной $S_1 = c_{vr} + c_{lp} + c_{ms}$, где

$$c_{vr} = \min\left\{c_{ir} \mid i \ne l, m\right\},\ c_{lp} = \min\left\{c_{lj} \mid j \ne r, s\right\}.$$

Пусть решение вспомогательной задачи является тройкой, в которую входит c_{lr} и не входит c_{ms}. Тогда оно доставляет сумму $S_2 = c_{lr} + c_{mp} + c_{vs}$, где

$$c_{mp} = \min\left\{c_{mj} \mid j \ne s, r\right\},\ \ c_{vs} = \min\left\{c_{is} \mid i \ne l, m\right\}.$$

Элементы решения вспомогательной задачи, не содержащем c_{lr} и c_{ms}, определяют величину $S_3 = \min\left\{c_{lp} + c_{mr} + c_{ws},\ c_{mq} + c_{ls} + c_{vr}\right\}$. Здесь (рис. 2)

$$c_{ws} = \min\left\{c_{is} \mid i \neq l, m\right\},\ c_{mq} = \min\left\{c_{mj} \mid j \neq s, r\right\}\ .$$

Определим величину, равную $MIN2 = \min\left\{S_1, S_2, S_3\right\}$. Ясно, что она соответствует искомой последовательности π_3, если $MIN2 \le MIN1$, иначе $\pi_3 = \left(\pi_3[l] = r,\right.$ $\left.\pi_3[m] = s,\ \pi_3[w] = q\right)$.

В общем случае последовательность $\pi_k = \left(\pi_k[i_1], \pi_k[i_2], ..., \pi_k[i_s], ..., \pi_k[i_k]\right)$ со свойствами а) и б) преобразуется в последовательность $\pi_{k+1} = \left(\pi_{k+1}[i_1],\ \pi_{k+1}[i_2], ...,\right.$ $\left.\pi_{k+1}[i_r], ...,\ \pi_{k+1}[i_{k+1}]\right)$ с этими же свойствами следующим образом.

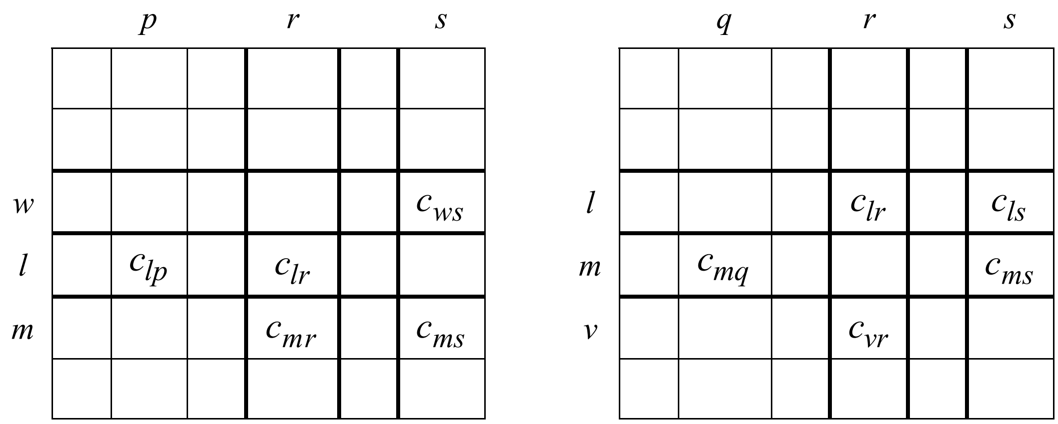


Рисунок 2

В матрице *C* находится

$$c_{\pi_k[i_{k+1}]} = \min\left\{c_{ij} \mid i \neq i_1, i_2, ..., i_s, ..., i_k,\ j \neq \pi_k[i_1], \pi_k[i_2], ..., \pi_k[i_s], ..., \pi_k[i_k]\right\},$$

формируется последовательность $\pi_{k+1}^1 = \left(\pi_k,\ \pi_k[i_{k+1}]\right)$ и вычисляется

$$MIN1 = \sum_{s=1}^{k} c_{\pi[i_s]} + c_{\pi_k[i_{k+1}]}.$$

Далее решается задача нахождения *k* + *1* элементов, которые в матрице *C* доставляют минимальную сумму своих значений и располагаются в разных строках и столбцах, включая все строки и столбцы с номерами, задаваемыми величинами $c_{\pi_k[i_1]}, c_{\pi_k[i_2]}, ..., c_{\pi_k[i_s]}, ..., c_{\pi_k[i_k]}$. Обозначим эту сумму *MIN2*. Найденные элементы образуют искомую последовательность $\pi_{k+1} = \pi_{k+1}^2$, если $MIN2 \le MIN1$. Иначе $\pi_{k+1} = \pi_{k+1}^1$.

Обоснование и описание рекурсивного алгоритма. Изложенная схема поиска оптимального назначения является основой алгоритма, в котором решение задачи (1) находится исключительно средствами теории паросочетаний для двудольных графов [1].

Матрице стоимостей ЗН *C* взаимно-однозначно соответствует двудольный граф (*X*, *Y*, *U*), где $|X| = |Y| = n$ и вершина $i \in X$ соединена с вершиной $j \in Y$ ребром $(i, j) \in U$ весом $c_{ij} \neq \infty$.

Паросочетание графа образует такое множество ребер, что никакие два из них не имеют общей вершины. Ребра, не входящие в паросочетание, называются свободными. Максимальное паросочетание – паросочетание с наибольшим числом ребер. Вершина, принадлежащая ребру паросочетания, называется насыщенной, остальные вершины графа – свободными. Ребро (i, j), включенное в паросочетание, обозначим $[i, j]$. Вершина j ребра $[i, j]$ определяется как напарник i. Совершенное паросочетание – это паросочетание, насыщающее все вершины графа. В двудольном графе (X, Y, U) мощность совершенного паросочетания, если оно существует, равна n. Решение ЗН σ состоит в построении в двудольном взвешенном графе (X, Y, U) совершенного паросочетания с минимальным суммарным весом ребер [1].

Пусть в графе зафиксировано паросочетание M. Простой путь называется чередующимся относительно паросочетания M, если рёбра пути через одно присутствуют в M [1]. Чередующийся путь, который начинается и заканчивается ребрами, не принадлежащими паросочетанию M, называется увеличивающим относительно паросочетания M. Следовательно, если $(i_1, j_2, i_2, j_3, ..., i_{k-1}, j_k, i_k, j_{k+1})$ – увеличивающий путь в двудольном графе (X, Y, U), то в нем свободны вершины i_1, j_{k+1} и $k+1$ рёбер (i_1, j_2), (i_2, j_3), ..., (i_{k-1}, j_k), (i_k, j_{k+1}). Остальные k ребер пути образуют паросочетание $\pi_k = ([i_2, j_2], [i_3, j_3], ..., [i_k, j_k])$. Длиной пути называется число встречающихся в нем ребер.

Предположим, что во взвешенном графе (X, Y, U), соответствующем матрице стоимостей ЗН C, не зафиксировано паросочетание. Тогда каждое ребро в (X, Y, U) является увеличивающим путем длины 1, а ребро с минимальным весом образует паросочетание или исходную последовательность π_1.

Пусть в графе (X, Y, U) построено паросочетание

$$\pi_k = ([i_1, j_1], [i_2, j_2], ..., [i_l, j_l], ..., [i_k, j_k])$$

с минимальной суммой $C(\pi_k)$ весов ребер среди всех паросочетаний мощности k. Преобразуем π_k в паросочетание $\pi_{k+1} = ([i'_1, j'_1], [i'_2, j'_2], ..., [i'_l, j'_l], ..., [i'_k, j'_k], [i'_{k+1}, j'_{k+1}])$, у которого величина $C(\pi_{k+1})$ достигает минимума на множестве Π_{k+1} всех паросочетаний мощности $k + 1$.

Паросочетание π_k разбивает множества X, Y соответственно на подмножества насыщенных вершин $I_k = \{i_1, i_2, ..., i_l, ..., i_k\}$, $J_k = \{j_1, j_2, ..., j_l, ..., j_k\}$ и на подмножестве свободных вершин $X - I_k = \{i_{k+1}, i_{k+2}, ..., i_s, ..., i_n\}$, $Y - J_k = \{j_{k+1}, j_{k+2}, ..., j_q, ..., j_n\}$.

Найдем свободное ребро весом

$$c_{i_r j_p} = \min\left\{c_{i_s j_q} \mid i_s \in X - I_k, j_q \in Y - J_k\right\}, \tag{2}$$

и присоединив его к паросочетанию π_k получим паросочетание $\pi'_{k+1} = \pi_k \cup [i_r, j_r]$ стоимостью $MIN1 = C(\pi_k) + c_{i_r j_p}$. Если π'_{k+1} не является паросочетанием π_{k+1} с минимальной суммой весов на множестве Π_{k+1}, то $\pi_{k+1} \in \Pi_{k+1} - \{\pi'_{k+1}\}$. Любое паросочетание в $\Pi_{k+1} - \{\pi'_{k+1}\}$ содержит l ребер π_k, $l = \overline{0, k-1}$ и $k - l + 1$ ребер увеличивающего пути относительно π_k.

Лемма. Пусть $\pi_{k+1} \in \Pi_{k+1} - \{\pi'_{k+1}\}$. Тогда $\pi_{k+1} = (P_{k+1} - \pi_k) \cup (\pi_k - P_{k+1})$, где P_{k+1} – кратчайший увеличивающий путь относительно паросочетания π_k.

Доказательство. Построим кратчайший увеличивающий путь P_{k+1} относительно паросочетания π_k, и, следовательно, паросочетание π^2_{k+1}. Покажем, что $\pi_{k+1} = \pi^2_{k+1}$. Ясно, что $C(\pi_k) \le C(\pi^2_{k+1})$. В величинах $C(\pi_k)$ и $C(\pi^2_{k+1})$ выделим общее слагаемое

$$C_0 = \sum_{\pi_k[i_s] \in \pi_k \cap \pi^2_{k+1}} c_{\pi_k[i_s]}.$$

Поэтому $C(\pi_k) - C_0 \le C(\pi^2_{k+1}) - C_0$. Но $\pi_k \cap \pi^2_{k+1} = \pi_k - P_{k+1}$, а $C(\pi^2_{k+1}) - C_0$ сумма весов ребер множества $P_{k+1} - \pi_k$. Эта сумма минимальна, когда P_{k+1} – кратчайший увеличивающий путь относительно паросочетания π_k. Следовательно, π^2_{k+1} – паросочетание с минимальной суммой весов ребер на множестве $\Pi_{k+1} - \{\pi'_{k+1}\}$, т.е. $\pi_{k+1} = \pi^2_{k+1}$.

Пусть вершина $j_l \in J_k$, $l \in \{1, 2, ..., k\}$ является концом свободного ребра (i_r, j_l) весом

$$c_{i_r j_l} = \min\left\{c_{i_s j_l} \mid i_s \in X - I_k\right\}, \tag{3}$$

вершина $i_f \in I_k, f \in \{1, 2, ..., k\}$ – началом свободного ребра (i_f, j_p) весом

$$c_{i_f j_p} = \min\left\{c_{i_f j_q} \mid j_q \in Y - J_k\right\}. \tag{4}$$

Обозначим X_k множество свободных вершин i_r, каждая из которых инцидентна ребру (i_r, j_l) весом, определяемым из (3), $|X_k| \le k$. Соответственно Y_k – множество свободных вершин j_p, инцидентных рёбрам (i_f, j_p) с весами, определяемыми из (4), $|Y_k| \le k$. Ясно, что кратчайший увеличивающий путь относительно паросочетания π_k начинается в вершине множества X_k и заканчивается в вершине множества Y_k.

Построим вспомогательный взвешенный орграф (V, A), обеспечивающий поиск кратчайшего увеличивающего пути P_{k+1} относительно паросочетания π_k. В орграфе (V, A) множество вершин $V = \{i_0\} \cup X_k \cup I_k \cup Y_k$.

Множество дуг A орграфа (V, A) представлено разбиением на подмножества A_0, A_1, A_2, A_3. Подмножество A_0 содержит $|X_k|$ дуг (i_0, i_r) нулевого веса, $i_r \in X_k$. В подмножество A_1 входит дуга $(i_r, i_l), i_r \in X_k, i_l \in I_k$, если, и только если, в паросочетании π_k вершина j_r ребра (i_r, j_l) – напарник вершины i_l. Дуге (i_r, i_l) присвоен вес $c(i_r, i_l) = c_{i_r j_l} + c_{i_l j_l}$. Дуга $(i_d, i_l) \in A_2$, $i_d, i_l \in I_k$, тогда и только тогда, когда в паросочетании π_k вершина j_l ребра $[i_d, j_l], j_l \in J_k$, является напарником i_l. Дуга (i_d, i_l) имеет вес $c(i_d, i_l) = c_{i_d j_l} + c_{i_l j_l}$. Подмножество A_3 включает дугу (i_f, j_p), $i_f \in I_k, j_p \in Y_k$, если вершины i_f и j_p соединены в графе (X, Y, U) ребром (i_f, j_p). Дуге (i_f, j_p) приписан вес $c(i_f, j_p) = c_{i_f j_p}$.

На рис. 3 представлен граф (*X*, *Y*, *U*), для которого получено паросочетание π_k, $k = 4$. Рёбра паросочетания изображены жирными линиями. Вершины i_5 и i_6 образуют множество X_k, а вершины j_5 и j_6 – множество Y_k. Рёбра (i_5, j_1), (i_6, j_2), (i_6, j_3) имеют вес, полученный из (3), а веса рёбер (i_2, j_5), (i_3, j_6), (i_4, j_6) определены из (4).

Вспомогательный орграф (V, A), построенный на графе (*X*, *Y*, *U*) и соответствующий паросочетанию π_4, изображен на рис. 4. Множество вершин *V* содержит вместе с вершиной i_0 подмножества $V_1 = \{i_5, i_6\}$, $V_2 = \{i_1, i_2, i_3, i_4\}$, $V_3 = \{i_7, i_8\}$. Множество дуг *A* образует подмножества

$$A_0 = \{(i_0, i_5), (i_0, i_6)\}, \; \mathrm{A}_1 = \{(i_5, i_1), (i_6, i_2), (i_6, i_3)\}, A_2 = \{(i_1, i_3), (i_1, i_4), (i_3, i_1)\},$$

$$\mathrm{A}_3 = \{(i_2, i_7), (i_3, i_8), (i_4, i_8)\}.$$

Из способа построения орграфа (*V*, *A*) следует, что множество путей, достижимых из вершины i_0 во все вершины $i_p \in V_3$, совпадает с множеством увеличивающих путей относительно паросочетания π_k, соединяющих в графе (*X*, *Y*, *U*) кардую вершину $i_r \in X - I$ с каждой вершиной $i_p \in Y - J$. Кроме того, любому пути орграфа (*V*, *A*), начинающемуся в вершине $i_r \in X - I$ в вершину $j_p \in Y - J$ относительно паросочетания π_k. Если в графе (*V*, *A*) построен кратчайший путь из вершины i_0 в вершину $i_p \in V_3$, то он содержит одну из дуг $(i_r, i_l) \in A_1$ и одну из дуг $(i_f, i_p) \in A_3$. Пусть $i_t \in V_3$ конечная вершина пути, кратчайшего среди всех путей из i_0 в остальные вершины множества V_3. Очевидно, такой путь в графе (*X*, *Y*, *U*) определит кратчайший увеличивающий путь P_{k+1} относительно π_k.

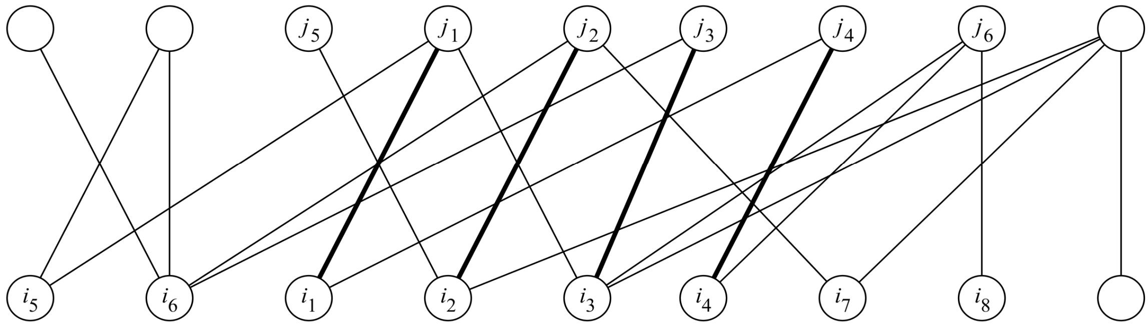


Рисунок 3

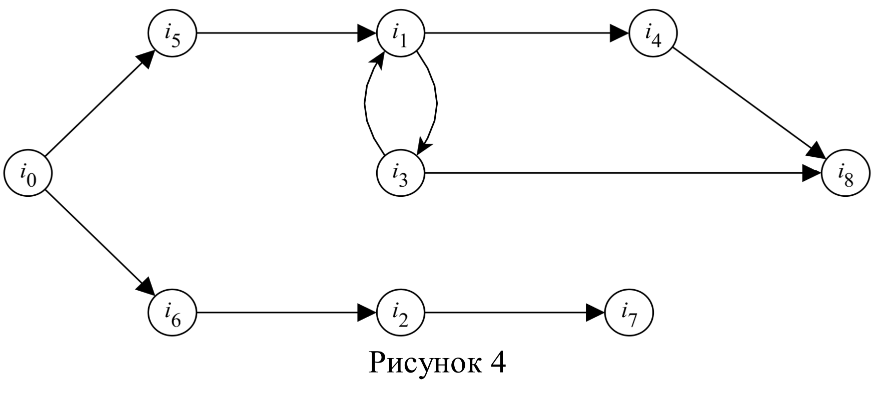


Рисунок 4

Таким образом, поиск в графе (X, Y, U) кратчайшего увеличивающего пути P_{k+1} относительно паросочетания π_k выполняется следующим образом.

1. При насыщенных вершинах $j_l \in J, i_f \in I, l, f \in \{1, 2, ..., k\}$, $|I| = |J| = k$, находятся свободные рёбра (i_r, j_l) и (i_f, j_p), $r, p \in \{k+1, k+2, ..., n\}$, с весами, определяемыми соответственно из (3) и (4).

2. Для множества всех свободных рёбер (i_r, j_l) и (i_f, j_p) и всех рёбер паросочетания π_k строится взвешенный орграф (V, A), предлагаемым выше способом.

3. В орграфе (V, A) ищется путь $(i_0, i_v, i_a, i_b, ..., i_e, i_d, i_t)$, кратчайший на множестве путей из i_0 в каждую вершину i_p, представленную в графе (X, Y, U) свободной вершиной j_p ребра (i_f, j_p).

Если существует такой путь, то существует кратчайший увеличивающий путь $P_{k+1} = (i_v, j_a, i_a, j_b, i_b, ..., i_e, i_d, i_d, i_t)$, $i_v \in X - I$, $j_a, j_b, ..., j_d \in J$, $i_a, i_b, ..., i_e$, $i_d \in I$, $j_t \in Y - J$, относительно паросочетания π_k, и следовательно, паросочетение π_{k+1}^2. Стоимость $C(\pi_{k+1}^2) = c_{i_v j_a} + c_{i_a j_b} + ... + c_{i_e j_d} + c_{i_d j_t}$. В противном случае не существует пути P_{k+1} и паросочетания π_{k+1}^2.

Очевидно, для нахождения в графе (X, Y, U) паросочетания π_{k+1}, стоимость которого минимальна на множестве π_{k+1} паросочетаний мощности $k+1$, достаточно

а) определить паросочетание $\pi'_{k+1} = \pi_k \cup [i_r, j_k]$ и суммарный вес её ребер $MIN1 = C(\pi_k) + c_{i_r j_p}$, π_k – паросочетание минимальной стоимости на множестве всех паросочетаний π_k мощности k, $c_{i_r j_p}$ – вес ребра $[i_r, j_p]$, определяемый из (2);

б) найти паросочетание π_{k+1}^2 и $MIN2 = C(\pi_{k+1}^2)$, построив кратчайший увеличивающий путь P_{k+1} относительно паросочетания π_k;

в) положить $\pi_{k+1} = \pi_{k+1}^1$, если $MIN1 \le MIN2$, и $\pi_{k+1} - \pi_{k+1}^2$ в противном случае.

Следующий алгоритм выполняет поиск решения ЗН, рекурсивно определяя в двудольном графе с $2n$ вершинами паросочетания π_k, содержащие k рёбер минимального суммарного веса.

Предлагаемый алгоритм состоит из такого же числа этапов и имеет такую же временную сложность, что и наилучший из известных методов оптимального назначения – венгерский метод [1].

S0. Алгоритм решения ЗН для матрицы стоимостей $C = [c_{ij}]_n$, $n \ge 2$, элементы которой принимают значения из множества неотрицательных действительных чисел или равны ∞; решение представлено совершенным паросочетанием $\pi = \pi_n$ с минимальной суммой $C(\pi)$ весов рёбер $[i, j]$ в двудольном графе (X, Y, U), $|X| = |Y| = n, i \in X, j \in Y$, где $c_{ij} \in R_0^+$, если $(i, j) \in U$, иначе $c_{ij} = \infty$; $k = 1$; найти $c_{i_k j_k} = \min\{c_{ij} \mid i, j = \overline{1, n}\}$, $I_k = \{i_k\}, J_k = \{j_k\}$, $\pi_k = \{[i_k, j_k]\}$, $C(\pi_k) = c_{i_k j_k}$.

S1. $k = k+1$; если $k > n$, то конец: построено решение ЗН π.

S2. Найти $c_{i_k j_k} = \min\left\{c_{ij} \mid i \in X - I_{k-1}, j \in Y - J_{k-1}\right\}$; если $c_{i_k j_k} = \infty$, то положить $MIN1 = \infty$, перейти к шагу S6, иначе $\pi_k^1 = \pi_{k-1} \cup \left[i_k, j_k\right], MIN1 = C\left(\pi_k^1\right)$.

S3. Найти все i_r такие, что для $j_l \in J_{k-1}, l \in \{1, 2, ..., k-1\}$, $c_{i_r j_l} = \min\left\{c_{ij_l} \mid i \in X - I_{k-1}\right\} \neq \infty$, и сформировать из них список X_k; если $X_k = \varnothing$, то положить $MIN2 = \infty$ и перейти к шагу S6.

S4. Найти все j_p такие, что для $i_l \in I_{k-1}, l \in \{1, 2, ..., k-1\}$, $c_{i_l j_p} = \min c\left\{_{i_l j} \mid j \in Y - J_{k-1}\right\} \neq \infty$, и сформировать из них список Y_k, если $Y_k = \varnothing$, то положить $MIN2 = \infty$ и перейти к шагу S6.

S5. Построить взвешенный орграф (V, A), $V = \{i_0\} \cup X_k \cup I_{k-1} \cup Y_k$ и выполнить в нем поиск пути, кратчайшего на множестве всех путей в вершины Y_k, достигаемые из i_0; если построен такой путь, то в графе (X, Y, U) найти соответствующий ему кратчайший путь

$$\pi_k^2 = \left(P_k - \pi_{k-1}\right) \cup \left(\pi_{k-1} - P_k\right),\ MIN2 = C\left(\pi_k^2\right),$$

иначе положить $\pi_k^2 = \varnothing, MIN2 = \infty.$

S6. Если $MIN1 = MIN2 = \infty$, то конец: не существует для матрицы $\left[c_{ij}\right]_n$ решения ЗН; Если $MIN1 \neq \infty$ или $MIN2 \neq \infty$, то если $MIN1 \leq MIN2$, то $\pi_k = \pi_k^1$, $I_k = I_{k-1} \cup \{i_k\}$, $J_k = J_{k-1} \cup \{j_k\}$,

иначе $\pi_k = \pi_k^2$, $I_k = \left\{i_l \mid l = \overline{1, k}; \left[i_l, j_l\right] \in \pi_k^2\right\}$, $J_k = \ = \left\{j_l \mid l = \overline{1, k}, \left[i_i, j_l\right] \in \pi_k^2\right\}$; перейти к шагу S1.

Теорема. Решение ЗН π корректно находится за время $O\left(n^3\right)$ построением в двудольном графе с $2n$ вершинами последовательностей паросочетаний π_k, $k = \overline{1, n}$, где π_k – паросочетание, содержащее k рёбер минимального веса, $\pi = \pi_k$.

Доказательство. Пусть построено паросочетание π_{k-1}, $k = \overline{2, n}.$ Алгоритм останавливается, когда 1) не находится ребра, объединение которого с π_{k-1} давало бы π_k^1; 2) во вспомогательном графе нет пути из любой вершины в любую вершину, следовательно, не существует увеличивающего пути относительно текущего паросочетания π_{k-1}.

Чтобы оценить трудоёмкость решения ЗН, заметим, что оно строится в результате выполнения n этапов, каждый из которых увеличивает паросочетание на одно ребро.

На первом этапе находится π_1 за время, оцениваемое в худшем случае величиной $O\left(n^2\right)$. На каждом следующем этапе строятся паросочетания π_k^1 и π_k^2, $k = \overline{2, n}.$ Время построения π_k^1 оценивается величиной $O\left(\left(n-k+1\right)^2\right)$. Нахождение π_k^2 требует $2\left(n-k+1\right)$ операций на поиск вершин множеств, k + 1 операций на построение

вспомогательного орграфа, $O\left(k^2\right)$ действий на построение в нем алгоритмом Дейкстры кратчайшего пути и $O(k)$ операций с множествами для определения π_k^2. Поэтому время каждого k-го этапа ограничено величиной $O\left(n^2\right)$. □

Пример. Исходная матрица $\left[c_{ij}\right]_n$ имеет вид:

	1	2	3	4	5	6	7
1	∞	∞	2	8	9	0	∞
2	15	∞	3	0	24	0	24
3	5	∞	∞	5	0	2	∞
4	∞	10	2	∞	23	∞	0
5	∞	0	0	∞	∞	∞	∞
6	0	15	10	3	17	∞	∞
7	14	0	24	2	2	15	∞

S0. $X = Y = \{1, 2, 3, 4, 5, 6, 7\}$, $k = 1$, $c_{i_1 j_1} = \min\left\{c_{ij} \mid i, j = \overline{1, 7}\right\} = c_{16} = 0$, $I_1 = \{1\}$, $J_1 = \{6\}$, $X_1 = Y_1 = \varnothing$, $\pi_1 = \{[1,6]\}$, $C(\pi_1) = c_{16} = 0$.

S1. $k = 2$.

S2. $c_{i_2 j_2} = \min\left\{c_{ij} \mid i \ne 1, j \ne 6\right\} = c_{24} = 0, \pi_2^1 = \pi_1 \cup [2,4] = \{[1,6],[2,4]\}$, $MIN1 = C\left(\pi_2^1\right) = c_{16} + c_{24} = 0 + 0 = 0.$

S3. Для $6 \in J_1$ $c_{i_r 6} = \min\{c_{i6} \mid i \ne 1\} = c_{26} = 0, X_2 = \{2\}$.

S4. Для $1 \in I_1$ $c_{1 j_p} = \min\left\{c_{1j} \mid j \ne 6\right\} = c_{13} = 2, Y_2 = \{3\}$.

S5. Взвешенный граф (V, A) представляет собой путь $(i_0, 2, 1, 3)$, где $c_{i_0 2} = 0$, $c_{21} = c_{26} + c_{16} = 0, c_{16} = 0, c_{13} = 2.$ Этому пути соответствует кратчайший увеличивающий путь $P_2 = (2, 6, 1, 3)$ относительно паросочетания $\pi_1 = \{(1, 6)\}$. Путь P_2 образуют рёбра $(2,6),(1,6),(1,3)$. Поэтому $\pi_2^2 = \{[2,6],[1,3]\}, MIN2 = c_{26} + c_{13} = 0 + 2 = 2.$

S6. Так как $MIN1 < MIN2$, $\pi_2 = \pi_2^1 = \{[1, 6], [2, 4]\}, I_2 = \{1, 2\}, J_2 = \{6, 4\}$.

S1. $k = 3$.

S2. $c_{i_3 j_3} = \min\left\{c_{ij} \mid i \ne 1, 2; j \ne 6, 4\right\} = c_{35} = 0,$ $\pi_3^1 = \pi_2 \cup [4, 7] = \{[1, 6], [2, 4], [3, 5]\}$, $MIN1 = C\left(\pi_3^1\right) = 0$.

S3. Так как $I_2 = \{1, 2\}, J_2 = \{6, 4\}$, то $c_{i_r 6} = \min\{c_{i6} \mid i \ne 1, 2\} = c_{36} = 2, c_{i_r 4} = \min\{c_{i4} \mid i \ne 1, 2\} = c_{74} = 2,$ и $X_3 = \{3, 7\}$.

S4. $c_{1 j_p} = \min\left\{c_{1j} \mid j \ne 6, 4\right\} = c_{13} = 2, c_{2 j_p} = \min\left\{c_{2j} \mid j \ne 6, 4\right\} = c_{23} = 3,\ Y_3 = \{3\}$.

S5. Подграфу двудольного графа (X, Y, U), определенному на множестве вершин $X_3 \cup I_2 \cup J_2 \cup Y_3$ (рис. 5) соответствует вспомогательный граф (рис. 6) с весами дуг: $c_{i_0 3} = c_{i_0 7} = 0,\ c_{31} = c_{36} + c_{16} = 2 + 0 = 2,\ c_{12} = c_{14} + c_{24} = 8 + 0 = 8,\ c_{21} = c_{26} + c_{16} = 0 + 0 = 0, c_{72} = c_{74} + c_{24} = 0 + 2 = 2,\ c_{23} = 3,\ 3 \in Y_3$.

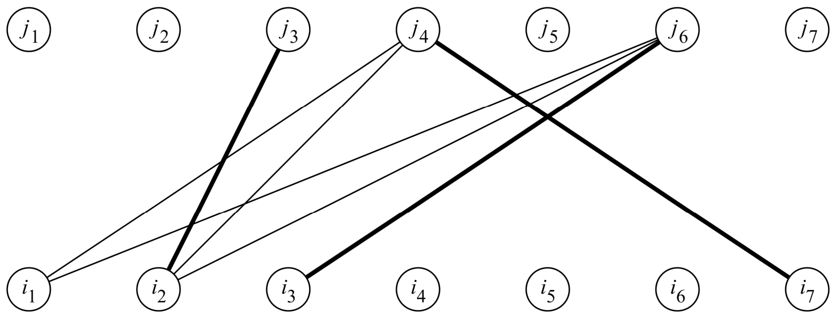


Рисунок 5

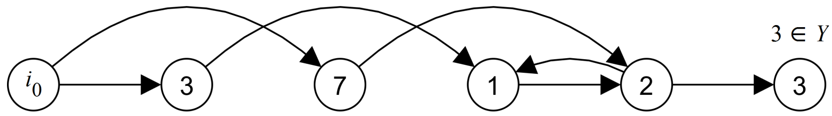


Рисунок 6

Кратчайший путь $(7, 2, 3)$, $3 \in Y$, связывающий в орграфе (V, A) вершины из X_3 с вершинами из Y_3, состоит из рёбер $(7, 4)$, $[2, 4]$, $(2, 3)$, $3 \in Y_3$, кратчайшего увеличивающего пути P_3 относительно паросочетания $\pi_2 = \{[1, 6], [2, 4]\}$.

Выполнив восемь итераций алгоритма, получим решение задачи:
$\pi_7 = \{[1, 6], [2, 4], [3, 5], [4, 7], [5, 3], [6, 1], [7, 2]\}$.

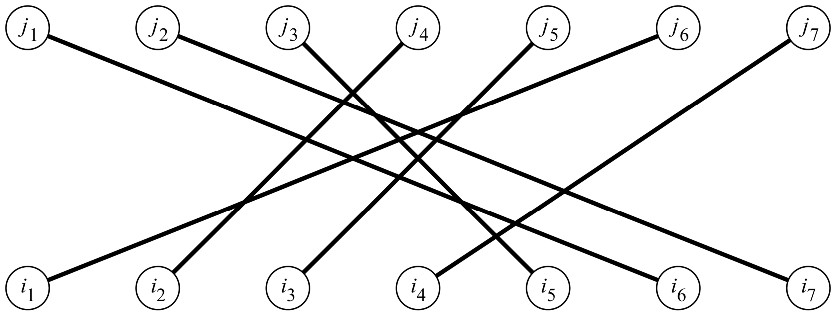


Рисунок 7

Для исследования времени решения задач разных размерностей различными методами проведен вычислительный эксперимент. Исследовалась зависимость времени решения ЗН от размерности входных данных. При помощи языка программирования Microsoft Visual C# были реализованы три метода решения ЗН: метод потенциалов, венгерский алгоритм и рекуррентный метод. На размерностях, больших 30, наилучшие результаты показал рекуррентный метод, а наихудшие – метод потенциалов.

Список литературы

1. Пападимитриу Х. Комбинаторная оптимизация: Алгоритмы и сложность / Х. Пападимитриу, К. Стайглиц. – М. : Мир, 1985. – 510 с.
2. Панишев А.В. Быстрый алгоритм решения задачи о назначениях для нахождения нижней границы стоимости маршрута коммивояжера / А.Ю. Левченко, А.В. Морозов, А.В. Панишев // Искусственный интеллект. Институт проблем искусственного интеллекта. – 2011. – С. 406-416.
3. Левченко А.Ю. Механизм ускорения вычислений в методе Литтла для решения задач класса коммивояжера / А.Ю. Левченко, А.В. Морозов, А.В. Панишев // Искусственный интеллект. – 2012. – С. 95-110.

References

1. Papadimitriou C. Combinatorial Optimization: Algorithms and Complexity / C. Papadimitriou, K. Steiglitz. – M.: Mir, 1985. – 510 p.
2. Panishev A.V. Fast algorithm for solving the assignment problem for finding lower bounds on the cost of the route of a Salesman / A.Y. Levchenko, A.V. Morozov, A.V. Panishev // Artificial intelligence The Institute of Artificial Intelligence.– Donetsk, 2011 .– P. 406-416.
3. Levchenko A.Y. Acceleration mechanism in the Little's method for solving problems in the class of a Salesman / A.Y. Levchenko, A.V. Morozov, A.V. Morozov // Artificial intelligence The Institute of Artificial Intelligence.– Donetsk, 2012. – P. 95-110.

RESUME

O.B. Matsiy, A.V. Morozov, A.V. Panishev

Recurrent Method for Solving the Assignment Problem

Well-known methods for solving the assignment problem, such as the Hungarian method, Huhn-Munkres method, method of potentials are constructed using different approaches in combinatorial optimization, and are characterized by different time complexity, no less than $O(n^3)$.

The paper proposes a new method for solving the assignment problem based on recursive obtaining an optimal solution. The assignment problem is formulated in the rearrangement matrix form that allows the use of a matrix approach to an optimal solution. The algorithm consists in finding a minimum total weight matching in the bipartite graph with $2n$ vertices.

Computational scheme of the recurrence method for solving the assignment problem is presented in a form adapted for implementation on a computer. The layout is the purpose of finding the optimal basis for the algorithm in which the solution is found solely by means of the theory of matchings for bipartite graphs presented in the rearrangement matrix form.

The developed method, in comparison with others, gives the opportunity to save computational resources, making a gain on large dimension of the input data.

Статья поступила в редакцию 11.03.2014.